

Un marco para la enseñanza de ingeniería distribuida de requisitos en las universidades

Gustavo Sevilla¹, Sergio Zapata¹, Estela Torres¹, Fáber Giraldo², Facundo Gallardo¹

¹Instituto de Informática / Facultad de Ciencias Exactas, Físicas y Naturales /
Universidad Nacional de San Juan;
{gsevilla,szapata,etorres,fgallardo}@iinfo.unsj.edu.ar

²SINFOCI / Universidad de Quindío, Armenia, Colombia; fdgiraldo@uniquindio.edu.co

RESUMEN

La colaboración y las tecnologías de comunicación son de fundamental importancia para alcanzar con éxito las metas comunes que se propone un equipo de trabajo global o virtual de desarrollo de software. Los problemas de distancia física, horaria y cultural afectan fuertemente aspectos sociales del equipo global, como la cohesión grupal, la confianza interpersonal, el intercambio de conocimiento, la interdependencia laboral, etc. No se han producido avances significativos en propuestas que mitiguen estos problemas dentro de un proceso de desarrollo de software global o distribuido.

Por otra parte la mayoría de los modelos de enseñanza actuales de Ingeniería de requisitos de Software (IRS) en las universidades no consideran estos nuevos escenarios distribuidos.

El presente proyecto intenta hacer aportes para atender la falencia mencionada, presentando un marco para la enseñanza de IRS distribuida en las universidades.

Palabras clave: Tecnología Aplicada en Educación, Análisis de Requisitos, Ingeniería de Requisitos Distribuidos, Experiencias Prácticas.

CONTEXTO

El presente trabajo esta inserto en el proyecto de investigación titulado “Colaboración, Confianza y Desarrollo Global de Software” (Código 21/E1061), financiado por la Universidad Nacional de San Juan (UNSJ) y cuyo desarrollo se realiza en el Instituto de

Informática de la Facultad de Ciencias Exactas, Físicas y Naturales.

1. INTRODUCCION

El proceso de desarrollo de software tiende, con mayor énfasis, a distribuirse globalmente donde los protagonistas de este proceso (partes interesadas) están dispersos geográficamente. Esta forma moderna de trabajo se llama Desarrollo Global de Software (DGS). Existen varias razones para este escenario: reducción de costos, mejor uso global de la escasa disponibilidad de recursos humanos, más horas de trabajo disponibles en relación con las diferentes zonas horarias, incentivos para invertir en mercados emergentes y un crecimiento significativo en la demanda de software en todo el mundo [1]. Es aceptable sostener que estos nuevos contextos de trabajo implicarán adaptaciones en las formas y contenidos de la enseñanza de Ingeniería de Software, buscando preparar profesionales para un contexto laboral globalizado.

Se sabe que el éxito de un proyecto de software depende en gran medida de una ejecución correcta del proceso de IRS. Tener una definición correcta de los requisitos del software es clave para obtener un producto de calidad que cumpla con las necesidades y expectativas del cliente / usuario. La necesidad de tener un proceso de IRS dentro del proceso de Ingeniería de Software es imperativa para obtener productos de calidad [2]. La IRS es fundamentalmente un proceso de comunicación entre un especialista en requisitos y el cliente [3].

La etapa de elicitación de requisitos es la más crítica de todas las fases en el desarrollo de software, ya que los errores cometidos en esta etapa son más costosos y difíciles de resolver debido a su impacto en las otras etapas [4], [5]. Asimismo, la calidad de los requisitos mejora en la medida en que hay una mayor participación de los interesados. Esta participación se dificulta cuando el proceso de IRS se lleva a cabo en un entorno distribuido de desarrollo, es decir, cuando los actores están geográficamente dispersos. De acuerdo con el trabajo de Damian [6], lograr la participación adecuada de los usuarios del sistema y el personal de campo es uno de los desafíos identificados en estos escenarios y se debe tanto a la comunicación inadecuada como a la diferencia horaria.

Lo expuesto justifica la necesidad de prestar atención a la enseñanza de la IRS en entornos de DGS. El modelo de enseñanza debe promover y facilitar la colaboración de los involucrados, siendo las técnicas de elicitación de requisitos y las herramientas de comunicación elementos clave para cumplir con este propósito.

La mayoría de los modelos de enseñanza actuales de IRS no consideran estos nuevos escenarios distribuidos. Además, rara vez alcanzan experiencia práctica profesional, y la enseñanza se centra frecuentemente en la teoría, mientras que los estudiantes rara vez se involucran en proyectos reales [7].

Este trabajo presenta un marco para la enseñanza de IRS distribuida. El marco aborda la ejecución de prácticas de IRS con equipos formados por estudiantes y profesores de universidades latinoamericanas. El modelo está respaldado por estrategias de colaboración basadas en Wiki, videoconferencia (VC) y correo electrónico. El modelo se aplica a las prácticas de elicitación de requisitos de software e incluye actividades de comunicación e interacción entre todos los participantes distribuidos de los proyectos de software (es decir, ingenieros de requisitos, desarrolladores, usuarios, etc.).

2. LINEAS DE INVESTIGACION Y DESARROLLO

La línea de investigación sobre la cual se ejecuta el presente trabajo intenta generar conocimiento científico sobre los aspectos sociales involucrados en equipos distribuidos o globales de desarrollo de software. Seaman [8] argumenta que la Ingeniería de Software es una compleja disciplina pues combina tanto aspectos técnicos como aspectos humanos.

Cada vez es más reconocida la importancia que tienen los vínculos sociales de los miembros de los equipos de desarrollo y su impacto en la efectividad del proceso de software. La complejidad de entender estas relaciones humanas se incrementa cuando los miembros de esos equipos están distribuidos y las distancias físicas, temporales, culturales, de lenguaje afectan las comunicaciones e interacciones de los mismos. Bajo este contexto de investigación es que desenvuelven las actividades del presente trabajo.

3. RESULTADOS OBTENIDOS/ESPERADOS

Al momento de llevar a cabo los trabajos experimentales descritos en esta sección, la enseñanza de la IRS en las carreras de Sistemas de la mayoría de las universidades latinoamericanas estaba enmarcada en escenarios colocalizados. Teniendo en cuenta esta situación descrita y la aparición de los nuevos escenarios de DGS, los experimentos consideraron la necesidad de investigar las técnicas de elicitación de requisitos y las herramientas de comunicación adecuadas para el proceso de ingeniería de requisitos en estos escenarios.

En el caso del primer experimento [9], se pretendía adquirir conocimientos de la etapa de elicitación distribuida de los requisitos de software en los nuevos escenarios de DGS, determinando las técnicas más convenientes de elicitación a utilizar. Se realizó un experimento controlado con dos factores: contexto de elicitación (distribuido/colocalizado) y técnicas de elicitación utilizadas (tres combinaciones diferentes de técnicas). Así, se

ejecutaron dos fases experimentales similares. La primera se aplicó a un contexto de desarrollo distribuido de software, mientras que la segunda se aplicó en un contexto colocalizado tradicional. El diseño experimental fue finalmente un diseño factorial 2x3.

Los resultados promedio del Nivel de Efectividad QRSD de cada combinación de técnica de elicitación, se pueden ver en las Tablas 1 y 2, agrupándolos por las técnicas de elicitación aplicadas en los dos procesos de elicitación realizados, en otras palabras, distribuidos y colocalizados, respectivamente.

Tabla 1. Promedio de resultados de la fase distribuida

Técnica	QRSD Promedio
Entrevista	66.76
Entrevista + Cuestionario	74.94
Entrevista + Brainstorming	68.94

Tabla 2. Promedio de resultados de la fase colocalizada

Técnica	QRSD Promedio
Entrevista	77.12
Entrevista + Cuestionario	77.54
Entrevista + Brainstorming	77.44

De los datos experimentales obtenidos surge que la combinación de técnicas de elicitación de requisitos más efectivas en un contexto distribuido sería la Entrevista-Cuestionario, que alcanza un nivel de efectividad que supera en el 9% la siguiente combinación de técnicas más efectiva, que fue la Entrevista-Brainstorming.

En el segundo experimento [10] alumnos avanzados hacían de Ingenieros de Requisitos e interactuaban con profesores que hacían de Clientes para obtener el Documento de Requisitos de Software (DRS) utilizando una Wiki en el proceso IRS en un escenario de desarrollo distribuido de software.

La Tabla 3 muestra los resultados obtenidos por cada grupo de IRS con respecto a tres atributos de calidad del Documento de requisitos de software: Completitud, Ausencia de ambigüedad y Precisión. Otro índice medido es el grado o nivel de uso de la herramienta wiki por cada grupo.

Tabla 3. Resultados Obtenidos

Grupo	Índice Uso de Wiki	Índice Completitud q1	Índice No-Ambiguo q2	Índice Precisión q3
G1	1.725	0.500	0.800	0.650
G2	1.529	0.540	0.941	0.765
G3	2.345	0.520	0.840	0.800
G4	2.300	0.460	0.700	0.750
G5	2.020	0.600	0.800	0.520
G6	1.618	0.500	0.618	0.382
G7	1.750	0.780	1.000	0.538
G8	1.339	0.520	0.731	0.346
G9	1.283	0.520	0.652	0.174

Sobre la base de la evidencia experimental obtenida y presentada anteriormente, respecto a las técnicas de elicitación de requisitos y las herramientas de comunicación utilizadas en los escenarios DGS, se propone un marco para la enseñanza de IRS en las universidades para escenarios de DGS donde el cliente/usuario está ubicado de forma remota con respecto al ingeniero de requisitos.

El marco propone, ver Figura 1, en base a las experiencias realizadas, el uso de técnicas de elicitación de requisitos y herramientas de comunicación en diferentes fases del proceso IRS. La propuesta trata de mantener una interacción continua con el cliente/usuario durante todo el proceso, teniendo el mismo diferentes grados de protagonismo en las diferentes fases. En este sentido, la wiki, específicamente configurada para el proceso IRS, es una herramienta de comunicación y repositorio que registra el progreso del proceso. Específicamente, los requisitos de software establecidos con sus comentarios asociados.

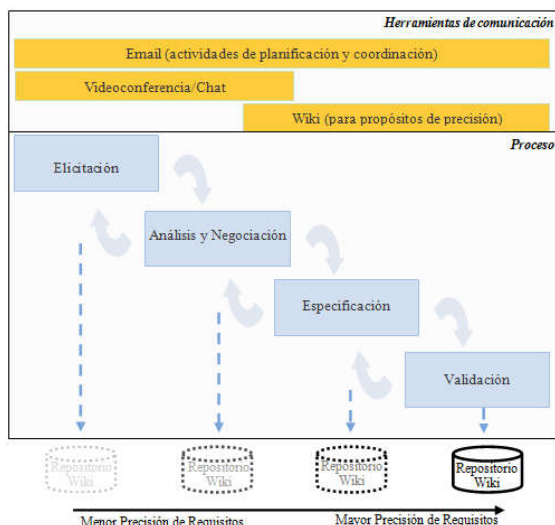


Figura 1: Marco propuesto

La Wiki es accedida tanto por el ingeniero de requisitos como por el cliente/usuario. La información que se genera en cada fase del proceso se almacena en la wiki que soporta el modelo de proceso.

Sobre la base de la evidencia experimental presentada en este documento, se propone que la comunicación asíncrona proporcionada por el correo electrónico se utilice durante todo el proceso de la IRS, en todas las fases. Normalmente, los correos electrónicos apoyan el intercambio de mensajes de planificación y coordinación entre las partes del proceso. Por ejemplo, solicitudes de VC, notificaciones de finalización de actividades, alarmas de tareas vencidas y difusión de aspectos relacionados con el progreso del proyecto, etc. Es por esta razón que el uso del correo electrónico aparece a lo largo de todo el proceso.

Al final del proceso, el conjunto de requisitos de software que formará el núcleo DRS del sistema permanecerá registrado en el Repositorio Wiki.

En el marco propuesto, el proceso de IRS se divide en cuatro fases diferentes:

-Fase de Elicitación: El marco establece que en esta primera fase del proceso, la técnica de la entrevista debe realizarse mediante VC. Este primer contacto entre el cliente/usuario y el ingeniero de requisitos apunta a obtener una definición general, no precisa, del problema a

resolver. Esta es la razón por la cual, como muestran los experimentos, las entrevistas no estructuradas son adecuadas para esta fase. Con este fin, la VC es una herramienta de comunicación de una riqueza comunicativa relativamente alta que soporta adecuadamente una entrevista remota. A medida que el conocimiento del problema progresa, se puede hacer el uso de cuestionarios para obtener una mayor precisión. Se pueden administrar de manera adecuada a través de una herramienta de comunicación asíncrona como el correo electrónico o de herramientas web de uso específico para cuestionarios como los formularios de Google.

-Fase de Análisis y Negociación: el marco establece que esta fase del proceso tiene como objetivo analizar la claridad, pertinencia y viabilidad técnica de los requisitos del software. Para este propósito, es necesario interactuar con el cliente/usuario para negociar aspectos relacionados con el esfuerzo y la viabilidad técnica de la implementación de los requisitos. En esta fase, las entrevistas deben realizarse a través de VCs. Como en todas las fases, los avances en términos de definición de requisitos de software se registran en el Repositorio Wiki. Al final de esta segunda fase es cuando la herramienta wiki como un medio de comunicación asíncrona entre el cliente/usuario y el ingeniero de requisitos comienza a tomar preponderancia.

-Fase de Especificación: esta fase del proceso tiene como objetivo establecer un conjunto de requisitos de software bien formulados, es decir, requisitos con un atributo de calidad suficiente para garantizar el éxito de las últimas etapas del proceso de desarrollo del software. El marco propone el uso de la herramienta wiki en esta fase, así el ingeniero de requisitos puede revisar y editar los requisitos para mejorar su calidad, mientras que el cliente/usuario puede verificar, modificar y validar los cambios. En esta fase, comienza un tipo de interacción asíncrona, textual, reflexiva y focal entre las partes, por lo que las VC ya no son útiles, convirtiendo a la wiki en la herramienta adecuada. Esta fase requiere un trabajo más detallado sobre los

requisitos, especialmente para darles precisión. Aquí es donde la herramienta wiki brinda un gran beneficio como lo demuestran los datos experimentales mencionados en este documento.

-Fase de Validación: Finalmente, en esta fase, ambas partes del proceso, cliente/usuario e ingeniero de requisitos, validan el conjunto de requisitos de software identificados y expuestos claramente en el Repositorio Wiki. Nuevamente, esta herramienta soporta estas tareas de revisión integral y validación.

La motivación adicional de los estudiantes de participar en estas prácticas junto con compañeros de otros países es un factor adicional significativo que contribuye positivamente al proceso de aprendizaje. El uso de herramientas de comunicación estándar, fácilmente disponibles en las universidades latinoamericanas, hace que el marco propuesto sea factible de implementar.

Resta aún realizar validaciones para efectuar afirmaciones sobre la utilidad del marco.

4. FORMACION DE RECURSOS HUMANOS

El equipo de trabajo es integrado por: Un Profesor Adjunto, un Profesor Titular, una Profesora JTP y un Ayudante alumno.

Esta línea de investigación permitió en el año 2018 obtener el título de Magister en Informática a uno de los autores de este artículo y a su vez constituye uno de los pilares para su estudio de doctorado próximo y una tesina de grado en desarrollo.

5. BIBLIOGRAFIA

[1] J. D. Herbsleb and D. Moitra, "Guest editors' introduction: Global software development," in *IEEE Softw.* 18, 2 (Mar. 2001), 16–20.

[2] A. Duran Toro, *Un entorno metodológico de ingeniería de requisitos para sistemas de información*, 2000.

[3] A. Abran, J. W. Moore, P. Bourque, R. Dupuis and L. L. Tripp, *Guide to the Software Engineering Body of Knowledge (SWEBOK) version 3 public review*. IEEE, 2013. ISO Technical Report ISO/IEC TR 19759.

[4] R. W. Ferguson and G. Lami, "An empirical study on the relationship between defective requirements and test failures," in *Software Engineering Workshop*, 2006. SEW'06. 30th Annual IEEE/NASA (2006), IEEE, pp. 7–10.

[5] P. Rajagopal, R. Lee, T. Ahlswede, C. C. Chiang and D. Karolak, "A new approach for software requirements elicitation," in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005. Sixth International Conference on (2005), IEEE, pp. 32–42.

[6] D. Damian & D. Zowghi, "RE challenges in multi-site software development organisations," in *Requirements Engineering* 8.3, 2003, 149-160.

[7] U. Bellur, "An academic perspective on globalization in the software industry," in *Computer Software and Applications Conference*, 2006, COMPSAC'06. 30th Annual International (2006), vol. 1, IEEE, pp. 53–54.

[8] C. B. Seaman, "Qualitative Methods". In: SHULL et al. (eds.), *Guide to Advanced Empirical Software Engineering*, Chapter 2, Springer, 2008.

[9] S. Zapata, E. Torres, G. Sevilla, L. Aballay and M. Reus, "Effectiveness of traditional software requirement elicitation techniques applied in distributed software development scenarios," in *CLEI 2012*, IEEE, pp. 1–7.

[10] G. Sevilla, S. Zapata, E. Torres and C. A. Collazos, "Using wikis as collaborative strategy to support software requirements elicitation," in *Computing Colombian Conference (9CCC)*, 2014 9th (2014), IEEE, pp. 54–61.